



ESZIO17 – Fundamentos de Processamento Gráfico

Laboratório 07: Mapeamento de Textura

1. Introdução

Nesta aula prática faz-se uma introdução aos os aspectos práticos de Mapeamento de Textura em objetos geométricos tridimensionais de computação gráfica através do OpenGL.

Acesse o MOODLE, e faça o *download* dos programas da aula que estão no arquivo “lab_MapTextur.zip”, e descompacte-o numa pasta.

2. Fundamentos iniciais

O mapeamento de textura (*texture mapping*) permite que voce cole uma imagem de uma parede de tijolos (por exemplo, obtido pelo scaneamento de uma foto real) num poligono e desenhar a parede inteira como sendo um único poligono. Por exemplo, quando a parede é vista em perspectiva, os tijolos podem se parecer mais pequenos à medida que a parede fica mais longe do ponto de visualização. Outros usos do mapeamento de textura incluem a pintura de vegetação em grandes poligonos representando o chão em simulação de voos; padroes de papel de parede; e texturas que fazem os poligonos se parecerem substancias naturais tais como marmore, madeira, ou tecido. As possibilidades sao infindáveis. Apesar de ser mais natural pensarmos em aplicar texturas sobre polígonos, as texturas também podem ser aplicadas em todas as primitivas – pontos, linhas, poligonos, bitmaps, imagens.

Material de consulta para esta aula: capítulo 9 do livro OpenGL Programming Guide (Red Book). Versão online: <http://www.glprogramming.com/red/chapter09.html>

3. Aplicando Mapeamento de Textura

Devido às muitas possibilidades, o mapeamento de textura é um assunto muito longo e complexo, e deve-se fazer várias escolhas de configuração durante sua programação quando for usa-la. Neste roteiro iremos mostrar apenas os fundamentos mais simples. Para mais detalhes recomendamos a leitura do capitulo 9 do livro “Red Book”.

Os passos para adicionar textura nos objetos de sua cena são:

1. Especificar a textura.
2. Indicar como a textura deve ser aplicada em cada pixel.
3. Habilitar o *texture mapping*.
4. Desenhar a cena, fornecendo ambas a textura e as coordenadas geometricas.

1º. Projeto: “checker.c” (verifique a execução e o funcionamento)

3.1 Criação de Objeto de Textura e Especificar uma Textura para este Objeto

Uma textura é usualmente considerada bi-dimensional, como a maioria das imagens, mas ela pode tambem ser uni-dimensional. O dado que descreve uma textura pode consistir de um, dois, tres, ou quatro elementos por texel, representando qualquer coisa a partir de uma

constante de modulação até uma quádrupla (R, G, B, A).

Neste primeiro exemplo, que é bem simples, um único objeto de textura é criado para manter uma única textura bi-dimensional.

A textura do tabuleiro xadrez é gerado na rotina **makeCheckImage()**, e toda a inicialização do mapeamento de textura ocorre na rotina **init()**.

glGenTextures() e **glBindTexture()** nomeia e cria um objeto de textura para uma imagens de textura. (Para mais informações, ver no capítulo 9 ["Texture Objects."](#))

O único mapa de textura, de resolução total, é especificado por **glTexImage2D()**, cujos parametros indicam o tamanho da imagem, o tipo da imagem, a localização da imagem, e outras propriedades. (Para mais informações, ver no capítulo 9 ["Specifying the Texture."](#))

3.2 Indicar como a Textura deve ser aplicada em cada Pixel

Voce pode escolher uma das quatro funcoes possiveis para calcular o valor RGBA final da cor fragmentada e do dado da imagem-textura. Uma possibilidade é simplesmente usar a cor de textura como sendo a cor final; isto é o modo “decal”, em que a textura é pintada no topo do fragmento, exatamente como um adesivo seria aplicado. (O primeiro exemplo usa o modo “decal”.) O modo “replace”, uma variacao do modo “decal”, é o segundo método. Outro método é o uso de textura para modular, ou escalonar, a cor do fragmento, esta técnica é util para combinar os efeitos de iluminação com texturização. Finalmente, uma cor constante pode ser misturada com aquela do fragmento, baseado no valor de textura.

As quatro chamadas para **glTexParameter*()** especificam como a textura deve ser envolvida e como as cores devem ser filtradas se nao existirem um ajuste exato entre os pixels da textura com os pixels da tela. (Para mais informações, ver no capítulo 9 ["Repeating and Clamping Textures"](#) e ["Filtering."](#))

3.3 Habilitando o Mapeamento de Textura

Voce necessita habilitar a texturização antes de desenhar sua cena. Texturização é habilitada ou desabilitada usando-se **glEnable()** ou **glDisable()** com a constante simbólica **GL_TEXTURE_1D** ou **GL_TEXTURE_2D** para a texturização uni- ou bi-dimensional, respectivamente. (Se ambas forem habilitadas, **GL_TEXTURE_2D** é a que sera usada.)

Na função **display()**, **glEnable()** habilita a texturização. O comando **glTexEnv*()** define o modo de textura para **GL_DECAL** tal que os poligonos texturizados são desenhados usando as cores do mapa de textura (ao invés de levar em consideracao que cor dos poligonos seriam desenhados sem textura).

3.4 Desenhando a Cena, fornecendo as coordenadas Geométricas e de Textura

É necessário indicar como a textura deve ser alinhada em relação aos fragmentos sobre os quais serão aplicados, antes de “gruda-los”. Isto é, voce necessita especificar tanto as coordenadas de textura quanto as geométricas à medida que especifica os objetos de cena. Para um mapa de textura bi-dimensional, por exemplo, as coordenadas de textura variam de 0.0 a 1.0 em ambas as direções, mas as coordenadas dos itens sendo texturizados podem ser qualquer coisa.

Para o exemplo da parede de tijolos, se a parede é quadrada e deseja-se representar uma copia da textura, o codigo provavelmente designaria as coordenadas de textura (0, 0), (1, 0), (1, 1), e (0, 1) para os quatro cantos da parede. Se a parede é muito grande, voce poderia

querer pintar varias copias do mapa de textura sobre ela. Se voce fizer isso, o mapa de textura deve ser designado tal que os tijolos na borda esquerda se ajustem graciosamente com os tijolos na borda direita, e similarmemente para os tijolos no topo com aqueles de baixo.

No exemplo, os dois poligonos são desenhados. Note que as coordenadas de textura são especificadas com as coordenadas dos vertices. O comando **glTexCoord*()** se comporta similarmemente ao commando **glNormal()**. **glTexCoord*()** define as coordenadas de textura corrente; qualquer comando de vértice subsequente possui aquela coordenada de textura associada com ela até que um novo comando **glTexCoord*()** seja chamado novamente.

4. Demais Atividades de Aula

2º. Projeto: “texbind.c”

- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
 - . Criação de duas texturas
 - . Desenho dos objetos no espaço 3D
 - . Designacao das texturas em cada objeto respectivo
- Responda:
 - . Qual o resultado da utilização de duas texturas?
 - . Como seria possível aplicar ambas texturas num mesmo objeto?

3º. Projeto: “textureRGB.c”

- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
 - . O carregamento da textura
 - . Definição das coordenadas de textura e de vertices
 - . Composição do objeto final da cena
 - . Rotação do objeto da cena
- Responda:
 - . O que é a textura utilizada?
 - . Porque foram definidas as normais?
 - . O que é o objeto desenhado?

5. Exercício para entregar

1) Modifique o exemplo do 3º. Projeto: “textureRGB.c” da seguinte forma:

- adicione a geração de duas texturas tal qual é feita no 2º projeto “texbind.c”, e
- utilize cada uma delas numa face diferente do objeto.

Importante: em todos os arquivos de código faça um cabeçalho (com comentários) colocando o seu nome completo, RA, data do programa, nome do programa, e exemplo de chamada do programa no prompt do Linux. Além disso, no **TÍTULO** das janelas OpenGL criadas pelo programa coloque o título do programa e o seu nome e sobrenome.

6. Relatório

Envie o relatório na forma de arquivo eletrônico HTML, e os programas C++, com cabeçalho (comentário) incluindo seu nome completo, RA, data do programa, nome do programa, e exemplo de chamada do programa no prompt do linux.

Elaborar um relatório contendo:

- Todos os procedimentos detalhados executados no laboratório.
- Respostas das perguntas
- Imagens obtidas. Tabelas de análises comparativas de resultados.
- Análise e conclusões.
- Exercícios para entregar